Natural Language Processing (NLP) – an introduction

https://www.ai2lab.ca/

Pedro Paiva
Postdoc Fellow Associate
Electrical and Software Engineering
Schulich School of Engineering

October 2024

Outline

- Context and Motivation
- Word Representation: word2vec & GloVe
- Modern Neural Networks: attention & transformer
- BERT: Bidirectional Encoder Representations from Transformers

Outline

- **Context and Motivation**
- **Word Representation: word2vec & GloVe**
- Modern Neural Networks: attention & transformer
- BERT: Bidirectional Encoder Representations from Transformers

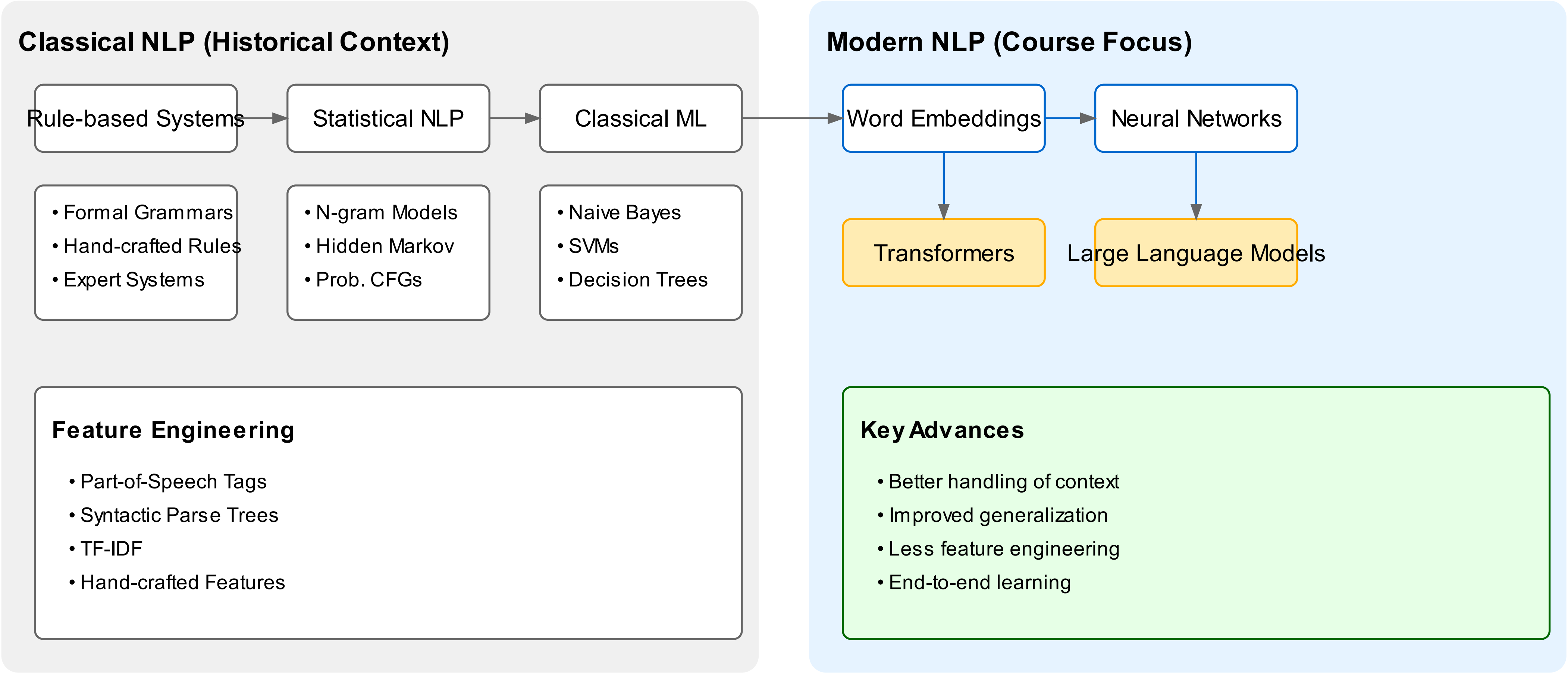
*DISCLAIMER !

- **What We Will Focus On:**
- Modern Word Representation Techniques:
  - Word2Vec, GloVe,
- Contextual embeddings
  - BERT
- Deep Learning Models for NLP
  - RNNs and Transformers
- Hands-on Learning
  - Practical applications

- **What We Will Not Cover:**
- Classical NLP Techniques:
  - Rule-based systems, syntactic parsing…
  - Statistical models
- Linguistic Theory
  - This lecture emphasizes data-driven, neural-based approaches

UNIVERSITY OF CALGARY

*DISCLAIMER !



Classical NLP (Historical Context)
Rule-based Systems
Statistical NLP
Classical ML
• Formal Grammars
• Hand-crafted Rules
• Expert Systems
• N-gram Models
• Hidden Markov
• Prob. CFGs
• Naive Bayes
• SVMs
• Decision Trees
Feature Engineering
• Part-of-Speech Tags
• Syntactic Parse Trees
• TF-IDF
• Hand-crafted Features
Modern NLP (Course Focus)
Word Embeddings
Neural Networks
Transformers
Large Language Models
Key Advances
• Better handling of context
• Improved generalization
• Less feature engineering
• End-to-end learning

Context and Motivation

Human language

- **Language Complexity**
  - Phonology (sound systems)
  - Morphology (word structure)
  - Syntax (sentence structure)
  - Semantics (meaning)
  - Pragmatics (context and use)

- **Language acquisition**
  - By age 3:
    - Has words for almost everything
    - Speaks three-word phrases.

- **Sociolinguistic aspects**
  - Dialects
  - Cultural influences

Fig 1. Language learning: training phase!

NLP Applications: the hype!

Chat Assistants / Conversational AI

- **Examples**
  - ChatGPT, Claude, LLAMMA, etc.

- **Key Features**
  - Natural language understanding and generation
  - Contextual awareness in conversations
  - Ability to perform various language tasks

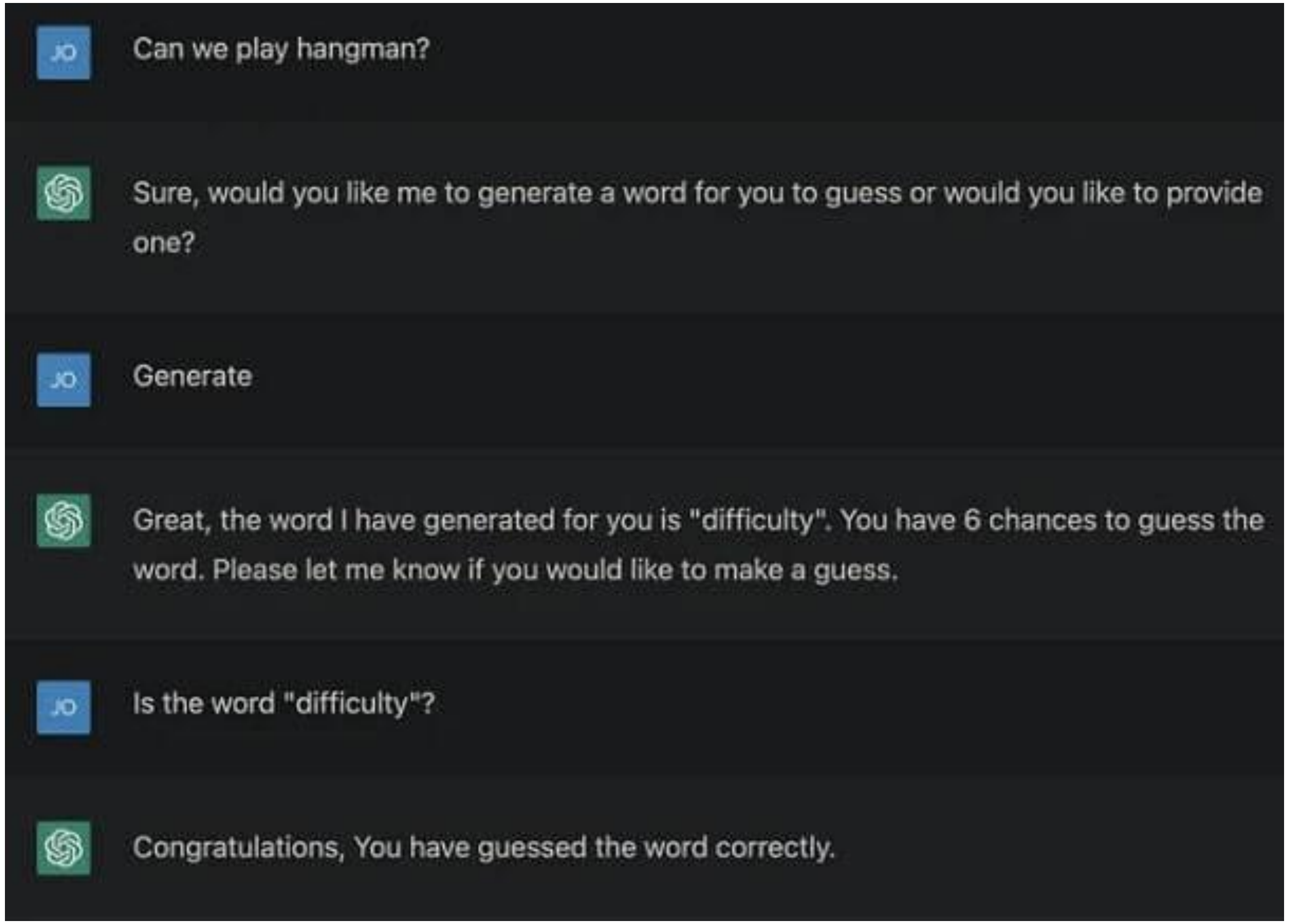


Fig 2. Conversation on OpenAI ChatGPT

NLP Applications: the hype!

Text-to-Image

Prompt

An expressive cat in the style of traditional ink wash painting, passionately playing a saxophone. The cat, wearing sunglasses, stands on its hind legs, fully immersed in the music. The fur is detailed with fluid brushstrokes, and the motion is captured with bold ink lines. The background is minimalistic, emphasizing the dynamic energy and intensity of the performance, with a focus on the cat's expressive posture and the flow of the ink --ar 3:4 --stylize 500 --v 6

Fig 3. Image generated with Midjourney

NLP Applications: the hype!

Reasoning

- A reasoning model is a computational system designed to simulate human-like reasoning. It uses logic, rules, and data to draw conclusions and make decisions.

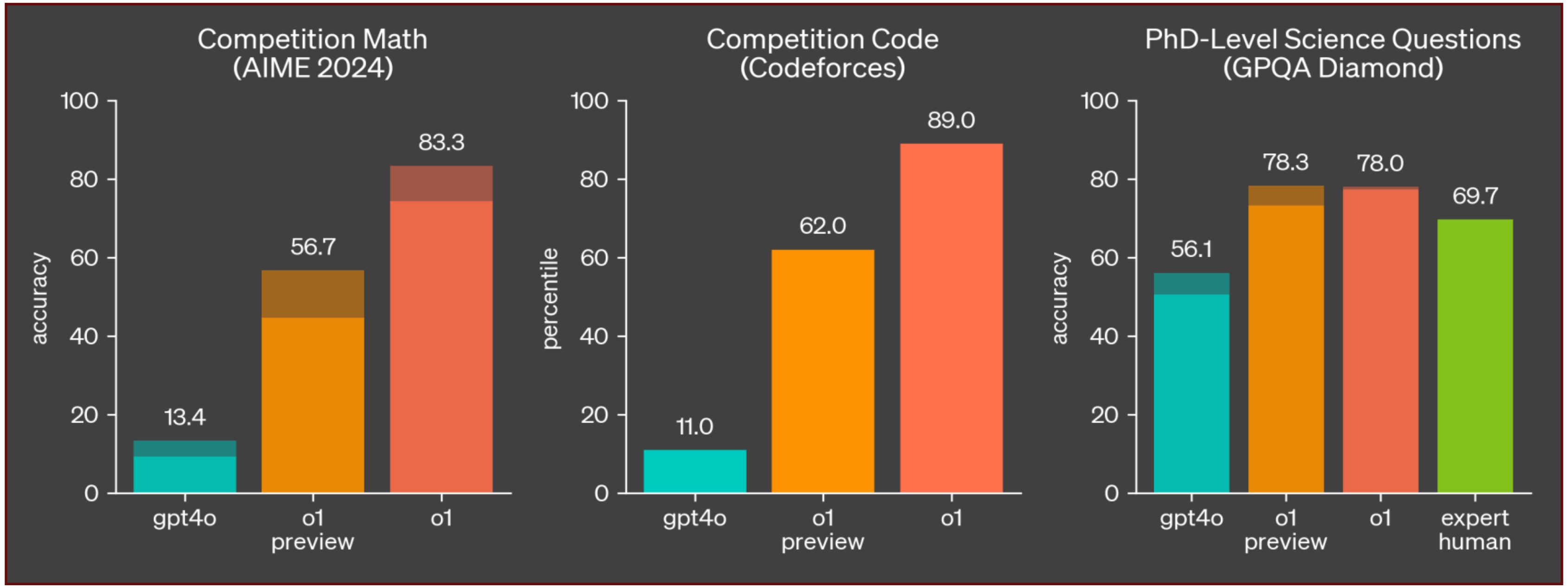


Fig 4. OpenAI o1 performance on a diverse set of human exams and ML benchmarks

Word Representation

Making computers understand words

Computers are excellent with numbers…

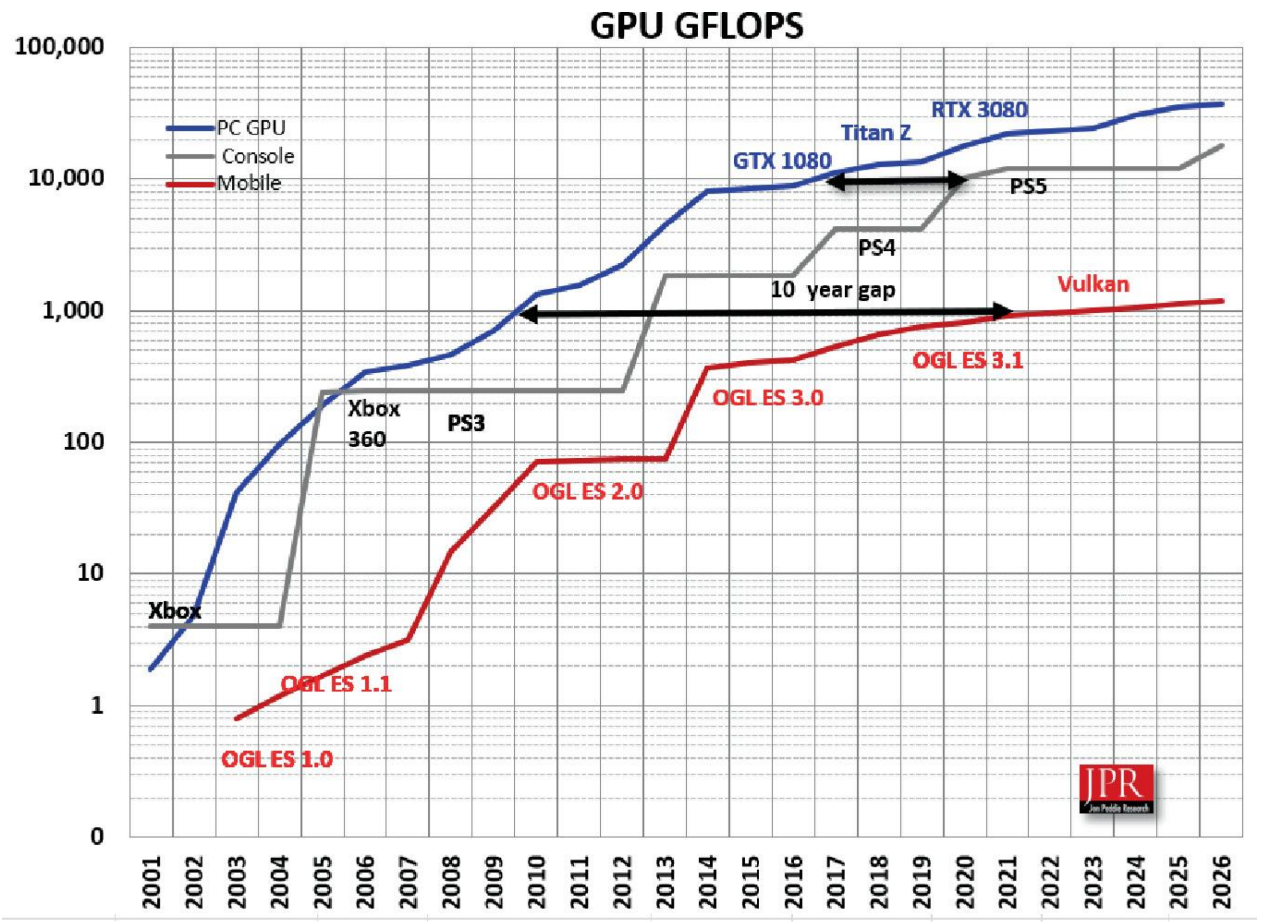


Fig 4. Comparison of GFLOPS of GPUs over time. Source: Peddie, J. (2023).

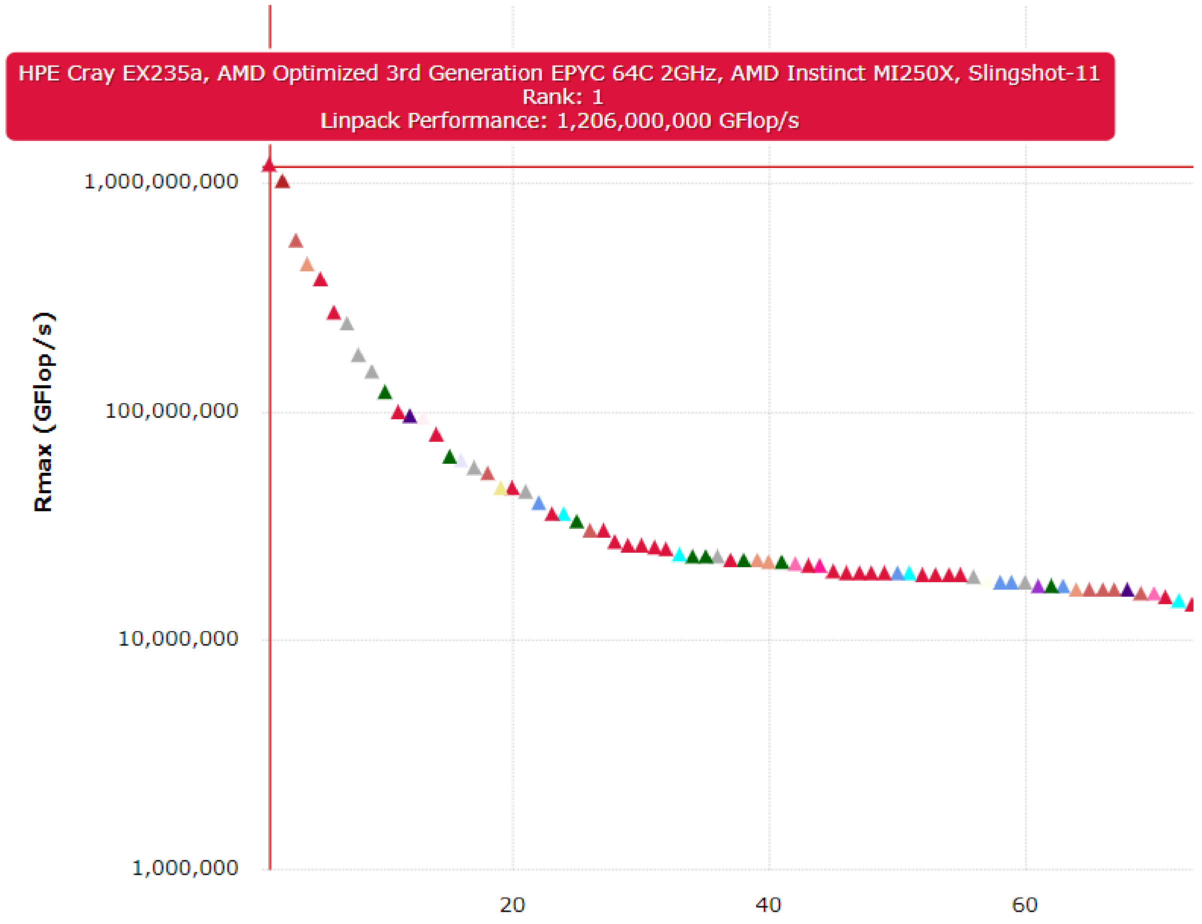


Fig 5. Top500 ranking on June/2024. Source Top500 List.

… not so much with words!

Homonyms

Synonyms

Context-Dependent Meaning

Idiomatic Expressions

Sarcasm

Sarcasm: “Great job!”

Tone and context can invert meaning. Computers often miss subtle cues humans use to detect sarcasm.

Representing words as numbers

Words	One-hot Vector
bat	[1, 0, 0, 0]
cat	[0, 1, 0, 0]
rat	[0, 0, 1, 0]
mat	[0, 0, 0, 1]

Problem Solved?

Cosine Similarity

v_{bat} = [1,0,0,0], ... , v_{mat} = [0,0,0,1]

$$v_i \cdot v_j = 0 \ (i \neq j)$$

$$sim \ \cos(v_i, v_j) = \frac{v_i \, . \, v_j}{\|v_i\| \|v_j\|} = 0$$

$$\frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \in [-1, 1]$$

Don’t forget the CONTEXT

Flowers bloom in the spring.

Context: **Time/Season**

Similar words: season, autumn, summer

The spring in the mattress is broken.

Context: **Mechanics**

Similar words: coil, bounce, elastic

We drank water from the natural spring.

Context: **Geography**

Similar words: fountain, source, wellspring

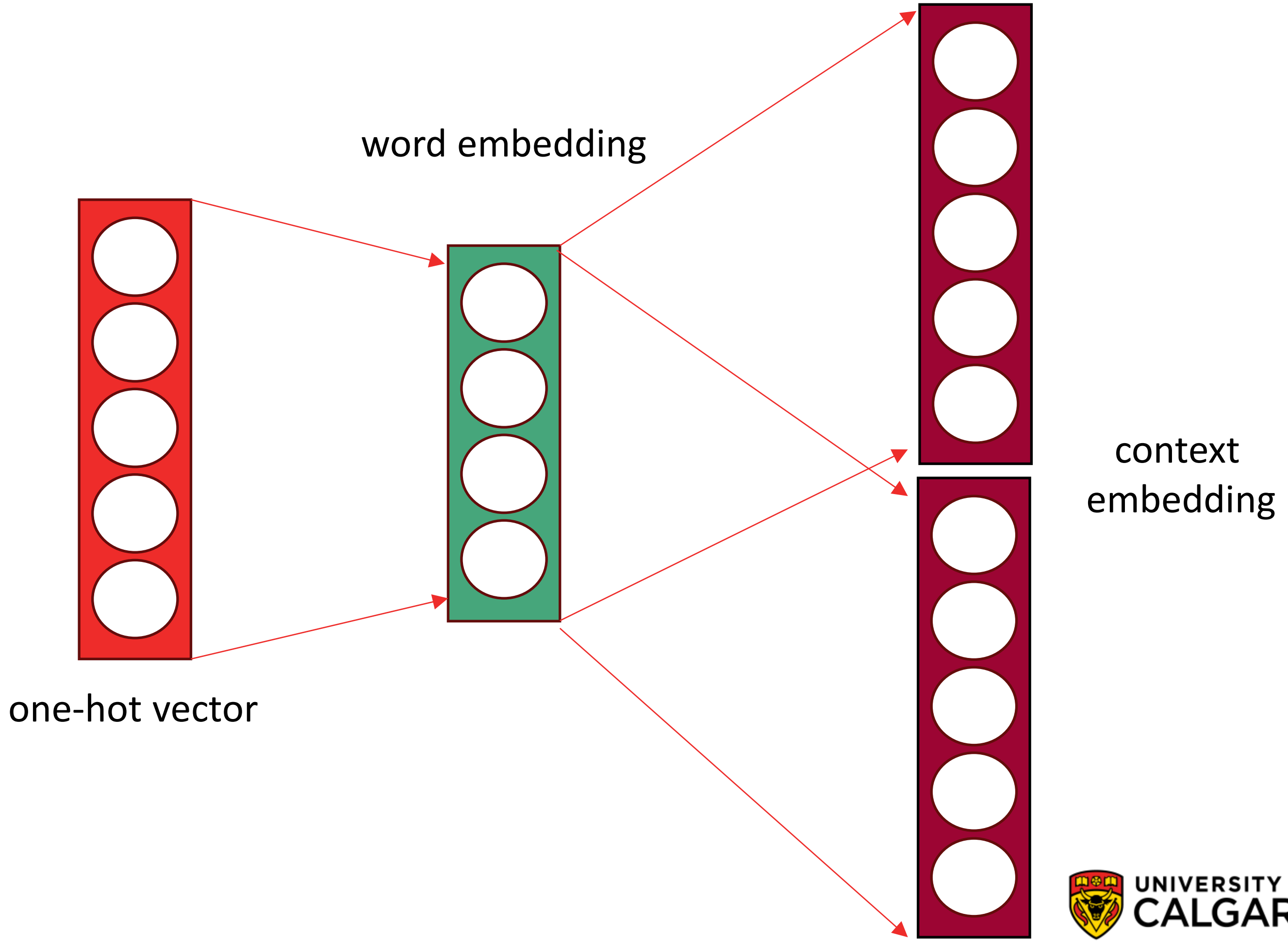
word2vec

* awesome visualization!

Self-supervised method to express word relationship using fixed length-vector and probabilities.

word2vec in a nutshell

1. Iterate over the vocabulary (corpus)
2. Predict the surrounding words
3. Take the gradient at the window



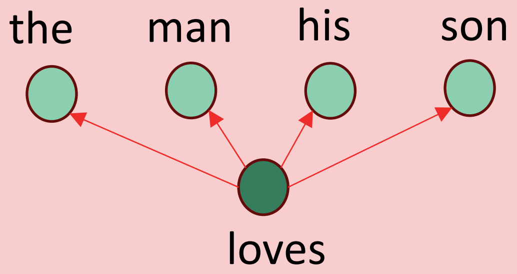
word2vec

[* awesome visualization!]

Self-supervised method to express word relationship using fixed length-vector and probabilities.

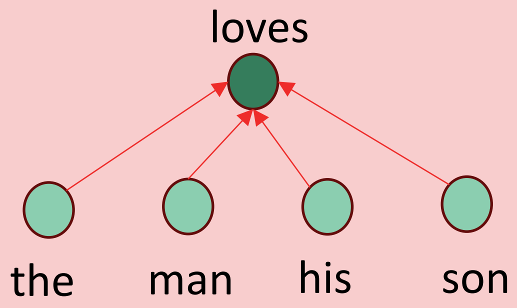
The Skip-Gram Models

$$P(w_o \mid w_c) = \frac{\exp\left(\mathbf{u}_o^\top \mathbf{v}_c\right)}{\sum_{i \in \mathcal{V}} \exp\left(\mathbf{u}_i^\top \mathbf{v}_c\right)}$$



Continuous Bag of Words

$$P(w_c \mid w_{o_1}, \ldots, w_{o_{2m}}) = \frac{\exp\left(\frac{1}{2m}\mathbf{u}_c^\top \left(\mathbf{v}_{o_1} + \ldots + \mathbf{v}_{o_{2m}}\right)\right)}{\sum_{i \in \mathcal{V}} \exp\left(\frac{1}{2m}\mathbf{u}_i^\top \left(\mathbf{v}_{o_1} + \ldots + \mathbf{v}_{o_{2m}}\right)\right)}$$



UNIVERSITY OF CALGARY

word2vec

The Skip-Gram Models

- Objective: given a word w_t predict its surrounding context words $w_{t-c}, \dots, w_{t+c}$ within a window size c

$P(w_{t-2} \mid w_t)$	$P(w_{t-1} \mid w_t)$		$P(w_{t+1} \mid w_t)$	$P(w_{t+2} \mid w_t)$		
The	**man**	**loves**	**his**	**son**	**very**	**much**
w_{t-2}	w_{t-1}	w_t	w_{t+1}	w_{t+2}		

word2vec

The Skip-Gram Models

- Objective: given a word w_t predict its surrounding context words $w_{t-c}, \dots, w_{t+c}$ within a window size c

	$P(w_{t-2} \mid w_t)$	$P(w_{t-1} \mid w_t)$		$P(w_{t+1} \mid w_t)$	$P(w_{t+2} \mid w_t)$	
The	**man**	**loves**	**his**	**son**	**very**	**much**
	w_{t-2}	w_{t-1}	w_t	w_{t+1}	w_{t+2}	

word2vec

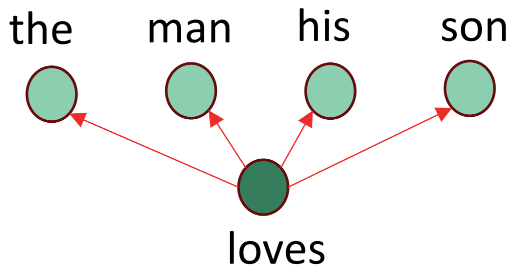
The Skip-Gram Models

- Objective: given a word w_t predict its surrounding context words $w_{t-c}, \ldots, w_{t+c}$ within a window size c

		$P(w_{t-2} \mid w_t)$	$P(w_{t-1} \mid w_t)$		$P(w_{t+1} \mid w_t)$	$P(w_{t+2} \mid w_t)$
The	**man**	**loves**	**his**	**son**	**very**	**much**
		w_{t-2}	w_{t-1}	w_t	w_{t+1}	w_{t+2}

word2vec

The Skip-Gram Models



P (“the”, “man”, “his” , “son” | “loves”)

P (“the” | “loves”) . P (“man” | “loves”) . P (“his” | “loves”) . P (“son” | “loves”)

Likelihood :

$$\prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} P\left(w^{(t+j)} \mid w^{(t)}\right)$$

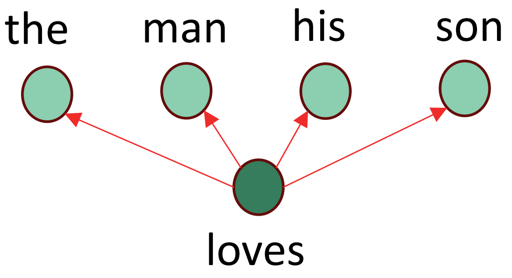
Loss function :

$$-\sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P\left(w^{(t+j)} \mid w^{(t)}\right)$$

- Sequence (length): $\boxed{\mathrm{T}} \rightarrow$ corpus / body of text
- Time step: t
- Context window: m

word2vec

The Skip-Gram Models



P (“the”, “man”, “his” , “son” | “loves”)

P (“the” | “loves”) . P (“man” | “loves”) . P (“his” | “loves”) . P (“son” | “loves”)

$$P\left(w_o \mid w_c\right)=\frac{\exp\left(\mathbf{u}_o^{\top} \mathbf{v}_c\right)}{\sum_{i \in \mathcal{V}} \exp\left(\mathbf{u}_i^{\top} \mathbf{v}_c\right)}$$

dot product!

- Vocabulary: $\mathcal{V}=\{0,1, \ldots,|\mathcal{V}|-1\}$
- Center word: $\boxed{\mathbf{u}_i}$
- Context word: $\boxed{\mathbf{v}_i}$

Vectors!

word2vec

Recap

dot product

- Measure similarity
- Thinking as vector space:
  - Point to the same direction if similar

$$u^\top v = u \cdot v = \sum_{i=1}^{n} u_i v_i$$

softmax

Exponentiate to make it positive

$$p_i = \frac{\boxed{e^{x_i}}}{\boxed{\sum_{j=1}^{N} e^{x_j}}}$$

Normalize to get probabilities

word2vec

The Skip-Gram Models

$$P\left(w_o \mid w_c\right)=\frac{\boxed{\exp\left(\mathbf{u}_o^{\top} \mathbf{v}_c\right)}}{\boxed{\sum_{i \in \mathcal{V}} \exp\left(\mathbf{u}_i^{\top} \mathbf{v}_c\right)}}$$

compare the similarity of *o* and *c*

normalize over the vocabulary

Loss function :

$$-\sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P\left(w^{(t+j)} \mid w^{(t)}\right)$$

$$\boxed{J(\theta)=-\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P\left(w^{(t+j)} \mid w^{(t)} ; \theta\right)}$$

word2vec

The Skip-Gram Models

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} \log P\left(w^{(t+j)} \mid w^{(t)}; \theta\right)$$

$$\theta = \begin{bmatrix} v_{aas} \\ v_{amaranth} \\ \vdots \\ v_{zoo} \\ u_{aas} \\ u_{ameise} \\ \vdots \\ u_{zoo} \end{bmatrix} \in \mathbb{R}^{2dV}$$

- How do we optimize the loss function for the whole vocabulary?

  Gradient of the function!

$$\nabla J(\theta)$$

word2vec

Recap

*extra content

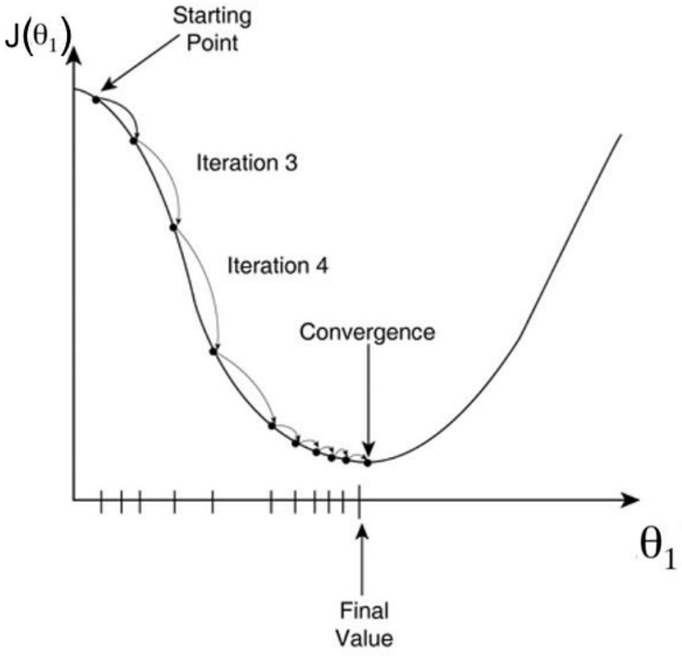
Gradient of a function

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]^\top$$

Gradient Descent

$f'(p) > 0 => f$ ☝️
$f'(p) < 0 => f$ 👇
$f'(p) = 0 => f$ 😑

Directional Derivatives



Cost Function – "One Half Mean Squared Error":

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

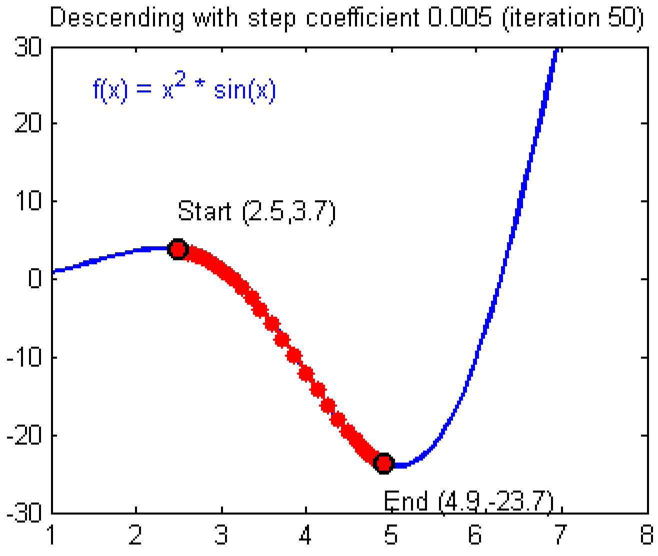
Objective:

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$$

Derivatives:

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x^{(i)}$$

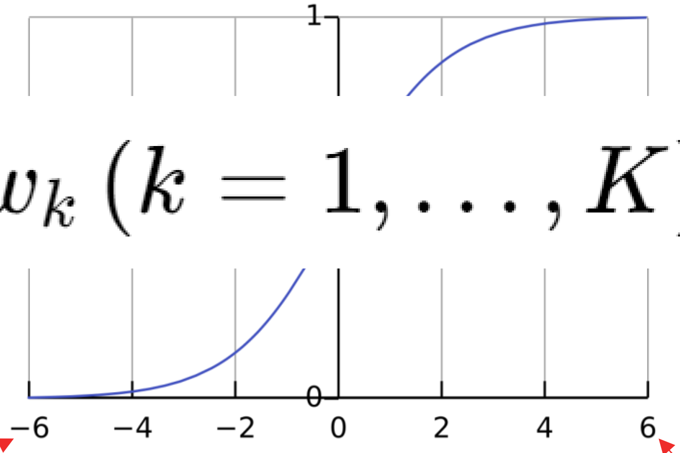


UNIVERSITY OF CALGARY

word2vec

Negative Sampling

Maximize words in the **same context** & Minimize the **same words** in **different contexts**



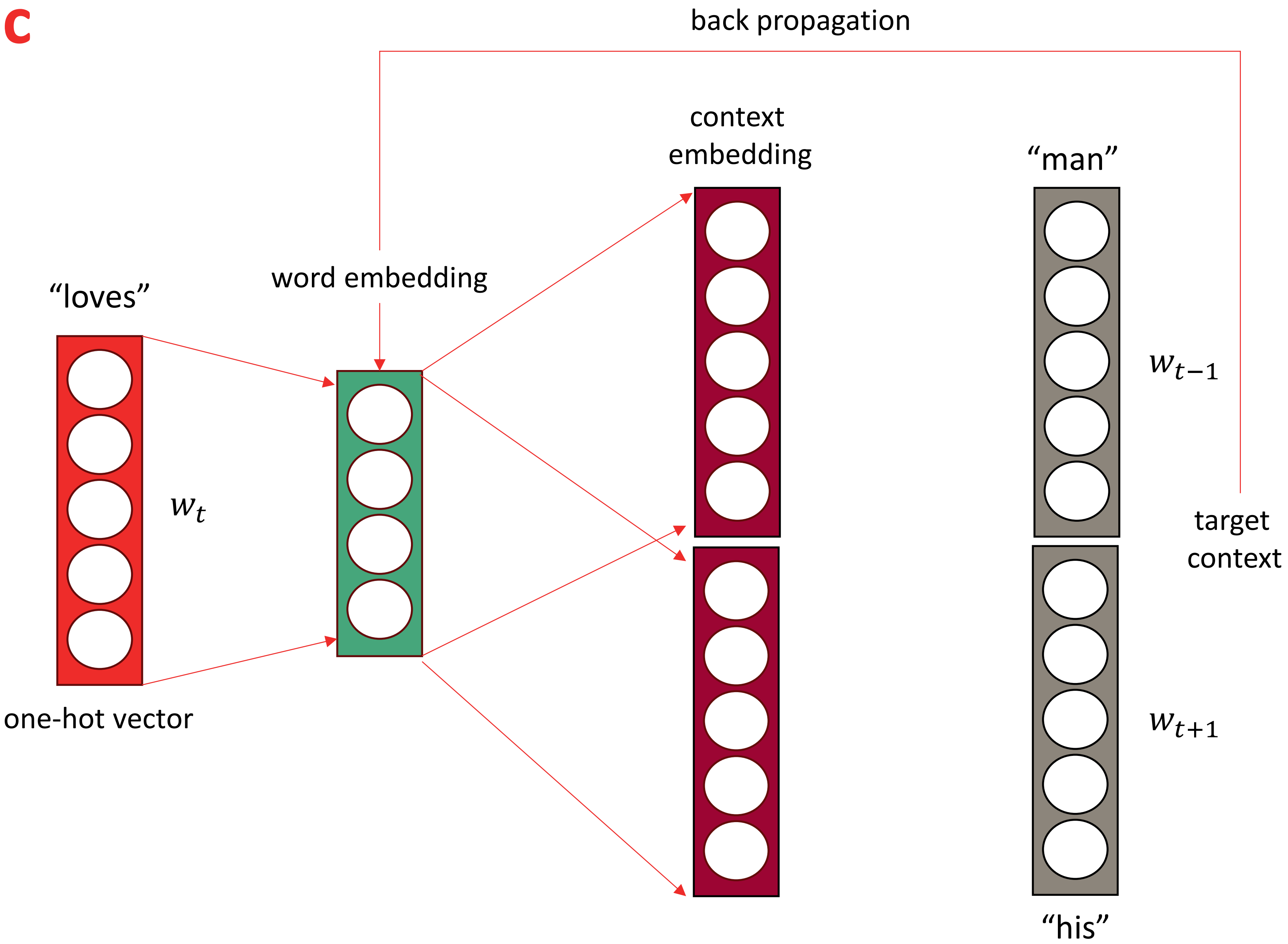
$$P(D = 1 \mid w_c, w_o) = \sigma(\mathbf{u}_o^\top \mathbf{v} \quad w_k \; (k = 1, \ldots, K)$$

Noise words
(out of context)

$$- \log \boxed{\sigma}\left(\mathbf{u}_{i_{t+j}}^\top \mathbf{v}_{i_t}\right) - \sum_{\boxed{k=1, w_k \sim P(w)}}^{K} \log \boxed{\sigma}\left(-\mathbf{u}_{h_k}^\top \mathbf{v}_{i_t}\right)$$

UNIVERSITY OF CALGARY

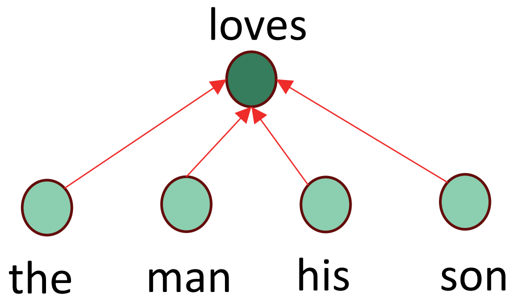
word2vec


back propagation
context
embedding
"man"
word embedding
"loves"
w_t
w_{t-1}
target
context
w_{t+1}
one-hot vector
"his"

word2vec

The Continuous Bag of Words (CBOW)

P (“love” | “the”, “man”, “his” , “son”)



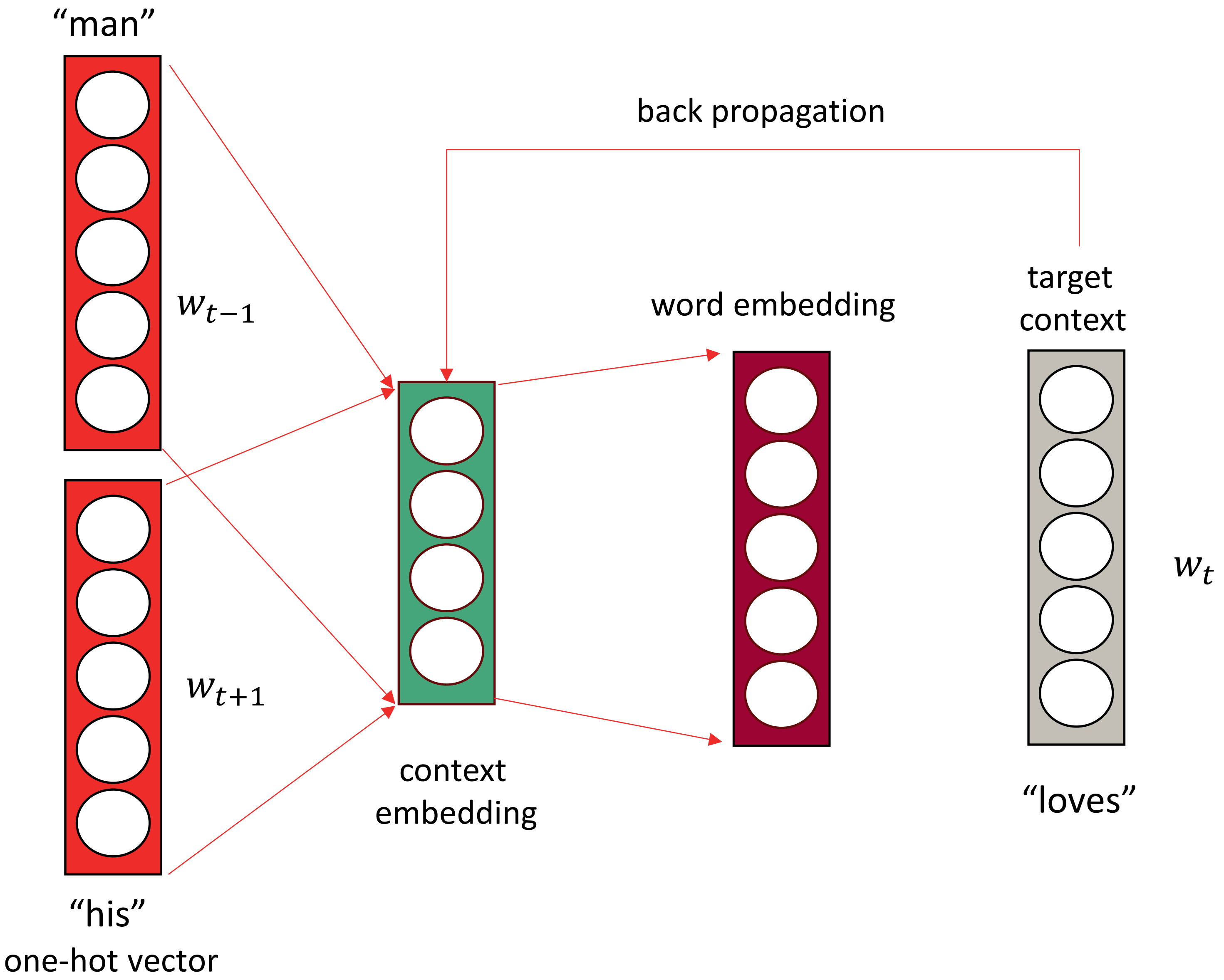
Likelihood :

$$\prod_{t=1}^{T} P\left(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots w^{(t+m)}\right)$$

Loss function :

$$-\sum_{t=1}^{T} \log P\left(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots w^{(t+m)}\right)$$

word2vec


“man”
back propagation
w_{t-1}
word embedding
target
context
w_t
w_{t+1}
context
embedding
“loves”
“his”
one-hot vector

word2vec

CBOW vs Skip-gram

CBOW	Skip-gram
PROS	PROS
1. Training is faster! 2. Low memory requirement 3. Good accuracy on frequent words	1. Works with small datasets 2. Can recognize rare occurrences
CONS	CONS
1. Requires large corpus	1. Memory/Process heavy*

*look on negative sampling

word2vec

Practice Time

Python

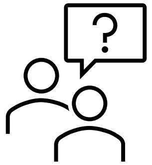
Keras

word2vec limitations

Why Do We Need More than Small Window Interactions?

Word2Vec Relies on Local Context!

- Skip-gram and CBOW **only use nearby words** (window size c) to learn embeddings.



- Important long-range dependencies are missed.

(e.g.) : "The man loves his **son** very much, even though they have not lived together for years, and despite the challenges that arose after his **son** moved to another country."

GloVe

GloVe: Global Vectors for Word Representation

… it **combines** the strengths of

- Global corpus statistics (**co-occurrence matrix**)
- **Dense embeddings** that capture relationships between words.

	the	man	loves	his	son	him	and	are	happy
the	0	2	0	0	0	0	0	0	0
man	2	0	1	0	0	0	0	0	0
loves	0	1	0	1	0	1	0	0	0
his	0	0	1	0	2	0	1	0	0
son	0	0	0	2	0	1	0	1	0
him	0	0	1	0	1	0	0	0	0
and	0	1	0	1	0	0	0	0	0
are	0	0	0	0	1	0	0	0	1
happy	0	0	0	0	0	0	0	1	0

vocabulary

1. "The man loves his son."
2. "His son loves him."
3. "The man and his son are happy."

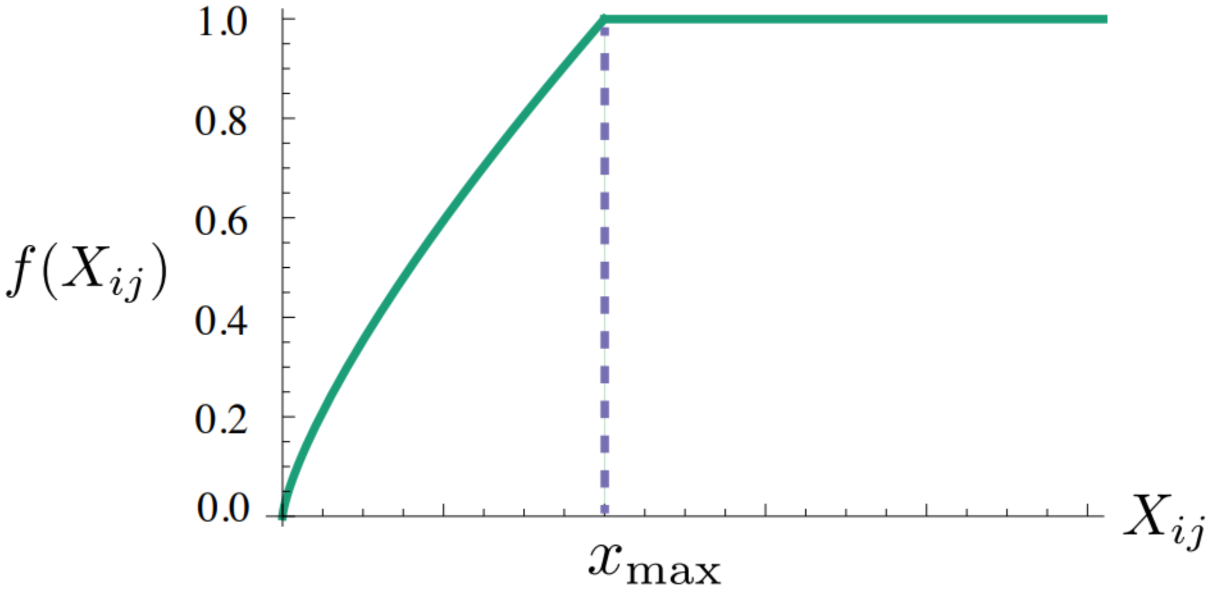
GloVe

- Counts alone are hard to interpret
  frequency doesn’t directly imply **importance** (e.g.: “the”)

- Global context isn’t captured

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	LARGE	LOW	LARGE	LOW
$P(k\|steam)$	LOW	LARGE	LARGE	LOW

Word-Word Co-occurrence << >> Probability Ratio



$$J = \sum_{i,j=1}^{V} \boxed{f(X_{i,j})}(w_i^T \tilde{w}_k + \boxed{b_i + \tilde{b}_j} - \log X_{ij})^2$$

common word importance reduction

bias reduction

GloVe

Key Strengths 💪

1. Efficiency
   1. One-time co-occurrence matrix construction
   2. Faster training than word2vec
2. Performance
   1. Strong on analogy tasks
   2. Better rare word representations
   3. Captures global corpus statistics

Practical Impact 🌟

1. Powers many modern NLP systems
2. Strong baseline for:
   1. Semantic similarity tasks
   2. Information retrieval
   3. Document classification
3. Foundation for contextual embeddings

In the next class…

- Context and Motivation
- Word Representation: word2vec & GloVe
- **Modern Neural Networks: attention & transformer**
- **BERT: Bidirectional Encoder Representations from Transformers**

Acknowledgments

ADVANCED IMAGING & ARTIFICIAL INTELLIGENCE LAB

https://www.ai2lab.ca/

References

1. Zhang, Aston, et al. "Dive into deep learning." *arXiv preprint arXiv:2106.11342* (2021).
2. Pennington, J., Socher, R., & Manning, C. D. (2014, October). **GloVe**: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
3. Mikolov, T. (2013). Efficient estimation of word representations in vector space (**word2vec**). *arXiv preprint arXiv:1301.3781*.

ADVANCED IMAGING & ARTIFICIAL INTELLIGENCE LAB

https://www.ai2lab.ca/

UNIVERSITY OF CALGARY